

**RELATÓRIO DO PROJETO**  
**GERENCIAMENTO DE ARQUIVOS**

Disciplina: Sistemas Operacionais  
Professor: Clóvis Ferraro  
Grupo: 07

## SUMÁRIO

|   |           |
|---|-----------|
| <b>1 INTRODUÇÃO .....</b>   | <b>3</b>  |
| <b>1.1 O que é um Sistema de Arquivos .....</b>                         | <b>3</b>  |
| <b>1.2 A importância do gerenciamento de arquivos e permissões.....</b> | <b>3</b>  |
| <b>1.3 Análise de arquivos .....</b>                                    | <b>3</b>  |
| <b>1.4 Análise de permissões .....</b>                                  | <b>4</b>  |
| <b>2 METODOLOGIA .....</b>  | <b>4</b>  |
| <b>2.1 Sistemas Operacionais utilizados. ....</b>                       | <b>4</b>  |
| <b>2.2 Ferramentas utilizadas .....</b>                                 | <b>4</b>  |
| <b>3 Análise dos Comandos e Configurações .....</b>                     | <b>5</b>  |
| <b>3.1 Linux Ubuntu 24.04.....</b>                                      | <b>5</b>  |
| <b>3.2 Windows 10 .....</b>   | <b>9</b>  |
| <b>3.3 Android .....</b>  | <b>13</b> |
| <b>4. Comparação e Análise Crítica.....</b>                             | <b>14</b> |
| <b>4.1 Linux .....</b>  | <b>14</b> |
| <b>4.2 Windows .....</b>  | <b>14</b> |
| <b>4.3 Android .....</b>  | <b>14</b> |
| <b>5 CONCLUSÃO .....</b>  | <b>15</b> |
| <b>6 AUTOAVALIAÇÃO.....</b>   | <b>15</b> |
| <b>7 REFERÊNCIAS.....</b>   | <b>16</b> |

## 1 INTRODUÇÃO

O objetivo deste relatório é implementar e analisar comandos de gerenciamento de arquivos e permissões em ambientes Windows, Linux e Android para compreender o controle de acesso e a estrutura dos sistemas de arquivos.

### 1.1 O que é um Sistema de Arquivos

Pense quando você faz o uso de um arquivo (Documentos, Planilhas, Apresentações ou até PDFs). E você salvar esse arquivo e precisa saber na onde que esse arquivo vai se localizar na memória. Isso que é o Sistemas de Arquivos, ele organiza seus arquivos de forma lógica para podermos interagir com eles como: diretórios, pastas, nomes dos arquivos e suas propriedades.

### 1.2 A importância do gerenciamento de arquivos e permissões

O gerenciamento de arquivos e permissões é essencial para a segurança, eficiência e organização dos dados. Ele pode proteger as informações contra acessos não autorizados e vazamentos, garantindo a conformidade com regulamentações e melhorando a facilidade em localizar documentos.

### 1.3 Análise de arquivos

A análise de arquivos será realizada nos principais sistemas operacionais utilizados: **NTFS** do Windows, **ext4** no Linux e **F2FS/ext4** no Android.

Cada sistema operacional possui características próprias em relação a estrutura de armazenamento.

### **1.4 Análise de permissões**

As permissões de arquivos são aplicadas através de mecanismos como Listas de Controle de Acesso (ACLs), que definem permissões de leitura, gravação e execução para usuários e grupos, e a criptografia que protege o conteúdo dos dados contra acessos não autorizados, mesmo que o arquivo seja roubado ou perdido.

## **2 MÉTODOLOGIA**

### **2.1 Sistemas Operacionais utilizados.**

Os Sistemas Operacionais que foram utilizados para a realização deste relatório.

**Windows 10, Linux Ubuntu 24.04 e Android x86**

### **2.2 Ferramentas utilizadas**

Ls -l, chmod, icacls, fsutil, du, df, createnew.

### 3 Análise dos Comandos e Configurações

#### 3.1 Linux Ubuntu 24.04

Usando o `ls -l`

Quando utilizamos o `ls -l` pode exibir as permissões dos arquivos.

Imagem 1 – `ls -l` no Ubuntu

```
ubuntu@ubuntu:~$ ls -l
total 0
drwxr-xr-x 2 ubuntu ubuntu 60 Oct 12 19:53 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Documents
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Downloads
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Music
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Public
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Videos
drwx----- 4 ubuntu ubuntu 80 Oct 12 19:54 snap
ubuntu@ubuntu:~$
```

Fonte: elaborado pelos autores, 2025.

Esse conjunto de 9 caracteres é o que define as permissões de arquivos e pastas

Imagem 2 – Explicação de diretório

`drwxr-xr-x`

Fonte: elaborado pelos autores, 2025.

Dando esse comando no terminal, podemos observar que o primeiro caractere começa com **d**, ou seja ele é um **diretório** ou uma **pasta**. O **-** (traço) representa a **permissão que está sendo negada**, e se for um **L** é um **link simbólico**.

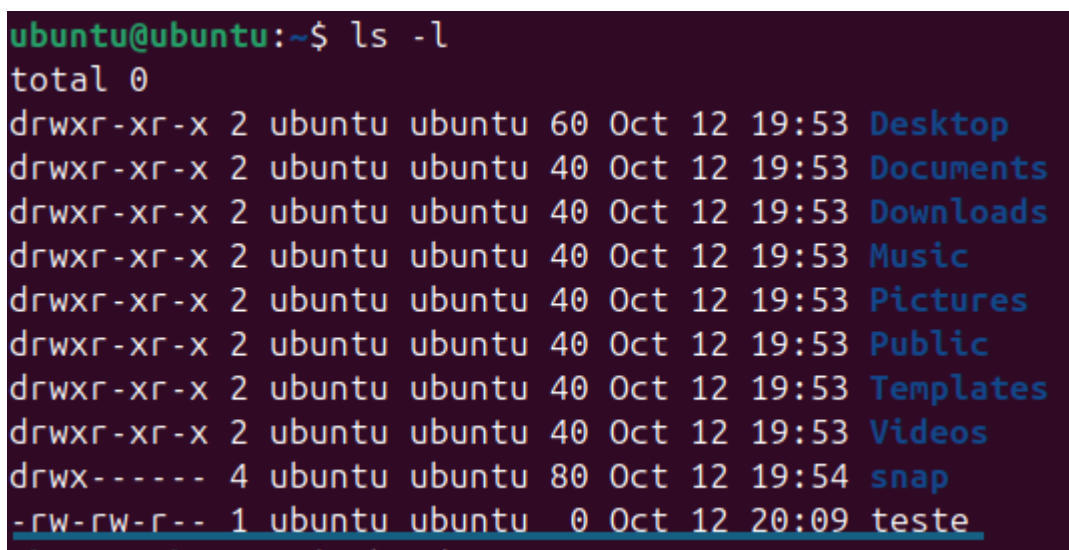
- R = Read – pode visualizar o conteúdo da pasta ou do arquivo;
- W = Write – pode modificar ou alterar arquivo/diretório;
- X = Execute – Um arquivo com essa letra é representado como um programa dentro dos sistemas Linux.

Cada três caracteres dessa seção representam as permissões das entidades diferentes num mesmo sistema.

Usando o **chmod**

O **chmod** vai realizar o modificamento das permissões dos arquivos, criamos um arquivo chamado teste, nele só podemos ler e escrever.

**Imagem 3** – Criação de arquivo.



```
ubuntu@ubuntu:~$ ls -l
total 0
drwxr-xr-x 2 ubuntu ubuntu 60 Oct 12 19:53 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Documents
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Downloads
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Music
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Public
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Videos
drwx----- 4 ubuntu ubuntu 80 Oct 12 19:54 snap
-rw-rw-r-- 1 ubuntu ubuntu  0 Oct 12 20:09 teste
```

**Fonte:** elaborado pelos autores, 2025.

Agora usando o **chmod** vamos alterar as permissões dessa pasta para ela poder executar o arquivo.

**Imagem 4** – Utilizando o chmod.

```

ubuntu@ubuntu:~$ chmod +x teste
ubuntu@ubuntu:~$ ls-l
ls-l: command not found
ubuntu@ubuntu:~$ ls -l
total 0
drwxr-xr-x 2 ubuntu ubuntu 60 Oct 12 19:53 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Documents
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Downloads
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Music
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Public
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Videos
drwx----- 4 ubuntu ubuntu 80 Oct 12 19:54 snap
-rwxrwxr-x 1 ubuntu ubuntu 0 Oct 12 20:09 teste

```

**Fonte:** elaborado pelos autores, 2025.

Vemos que agora foi incluído o **x** de Execute, ou seja agora o usuário tem a permissão de executar esse arquivo.

E se quisermos tirar uma permissão? Vamos ver a seguir.

**Imagem 5** – Utilizando o chmod

```

ubuntu@ubuntu:~$ chmod -w teste
ubuntu@ubuntu:~$ ls-l
ls-l: command not found
ubuntu@ubuntu:~$ ls -l
total 0
drwxr-xr-x 2 ubuntu ubuntu 60 Oct 12 19:53 Desktop
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Documents
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Downloads
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Music
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Pictures
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Public
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Templates
drwxr-xr-x 2 ubuntu ubuntu 40 Oct 12 19:53 Videos
drwx----- 4 ubuntu ubuntu 80 Oct 12 19:54 snap
-r-xr-xr-x 1 ubuntu ubuntu 0 Oct 12 20:09 teste

```

**Fonte:** elaborado pelos autores, 2025.

Nesse caso foi retirada a permissão de escrever no arquivo, o usuário não poderá modificar esse arquivo, somente executar e ler.

### Comandos **df** e **du**

Os comandos **df** e **du** no Linux são usados para verificar o uso do disco, mas com propósitos diferentes: **df** (disk free) mostra o espaço total disponível em sistemas de arquivos montados, enquanto **du** (disk usage) exibe o espaço ocupado por arquivos e diretórios específicos. O **du** é mais útil para encontrar o que está ocupando espaço dentro de um diretório, e o **df** é bom para ter uma visão geral do sistema de arquivos.



## 3.2 Windows 10

### Utilizando o **icacs**

O **icacs** é um comando pela CLI que permite exibir e gerenciar permissões de acesso (ACLs) de arquivos e pastas de forma mais eficiente do que pela GUI.

Imagem 6 – Utilizando o **icacs**

CA. Administrador: Prompt de Comando

```
Microsoft Windows [versão 10.0.19045.6332]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>icacs.exe

ICACLS name /save aclfile [/T] [/C] [/L] [/Q]
    armazena as DACLS referentes aos arquivos e às pastas correspondentes
    a name em arquivo_ACL para uso posterior com /restore. Observe que
    as SACLs, o proprietário e os rótulos de integridade não são salvos.

ICACLS directory [/substitute SidOld SidNew [...]] /restore aclfile
    [/C] [/L] [/Q]
    aplica as DACLS armazenadas aos arquivos de diretório.

ICACLS name /setowner user [/T] [/C] [/L] [/Q]
    altera o proprietário de todos os nomes correspondentes. Essa opção
    não força uma alteração de propriedade; para essa finalidade, use
    o utilitário takeown.exe.

ICACLS name /findsid Sid [/T] [/C] [/L] [/Q]
    localiza todos os nomes correspondentes que contêm uma ACL
    com menção explícita a Sid.

ICACLS name /verify [/T] [/C] [/L] [/Q]
    localiza todos os arquivos cuja ACL não está na forma canônica
    ou cujo tamanho é inconsistente com as contagens de ACEs.

ICACLS name /reset [/T] [/C] [/L] [/Q]
    substitui as ACLs por ACLs herdadas padrão para todos os arquivos
    correspondentes.

ICACLS name [/grant[:r] Sid:perm[...]]
    [/deny Sid:perm [...]]
    [/remove[:g|:d] Sid[...]] [/T] [/C] [/L] [/Q]
    [/setintegritylevel Level:policy[...]]

    /grant[:r] Sid:perm concede direitos de acesso ao usuário especificado.
    Com :r, as permissões substituem todas as permissões explícitas
    concedidas anteriormente. Sem :r, as permissões são adicionadas
    às permissões explícitas concedidas anteriormente.

    /deny Sid:perm nega explicitamente direitos de acesso ao usuário
    especificado. Uma ACE de negação explícita é adicionada para as
    permissões declaradas e as mesmas permissões em concessões explícitas
```

Fonte: elaborado pelos autores, 2025.

### Suas principais funcionalidades

- Listar permissões
- Modificar permissões
- Salvar e restaurar permissões
- Gerenciar proprietários

### Letras que representam o comando:

- F – Controle total
- M – Modificar
- RX – Ler e Executar
- R – Ler
- W – Gravar

Modificadores do **icacls** são opções de linha de comando que controlam como o comando se comporta, permitindo listar, alterar, adicionar ou remover permissões em arquivos e pastas do Windows de forma eficiente.

### Principais modificadores:

- /grant ou deny – Concede ou nega a permissão de acesso a um usuário.
- Sid: perm: - Define as permissões F,M,R para um determinado Sid (Identificador de Segurança).

**Imagem 7** – Utilizando o icacls

```
C:\Windows\system32>icacls c:\intel
c:\intel BUILTIN\Administradores:(I)(OI)(CI)(F)
        AUTORIDADE NT\SISTEMA:(I)(OI)(CI)(F)
        BUILTIN\Usuários:(I)(OI)(CI)(RX)
        AUTORIDADE NT\Usuários autenticados:(I)(M)
        AUTORIDADE NT\Usuários autenticados:(I)(OI)(CI)(IO)(M)

Processados com sucesso 1 arquivos; falha no processamento de 0 arquivos
```

**Fonte:** elaborado pelos autores, 2025.

Os termos (OI) e (CI) na pergunta se referem a modificadores de herança: (OI) (Object Inherit) significa que a permissão será herdada por objetos dentro do diretório, e (CI) (Container Inherit) significa que a permissão será herdada por outros contêineres (pastas) dentro desse diretório.

Utilizando o **fsutil**.

**Fsutil** é um comando no Windows usado para realizar tarefas avançadas de gerenciamento de sistemas de arquivos, como criar links simbólicos, gerenciar cotas de disco, criar e gerenciar arquivos e consultar informações do sistema de arquivos.

**Imagem 8** – Utilizando o fsutil

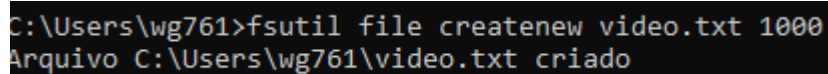
```
C:\Windows\system32>fsutil
---- Comandos com suporte ----

8dot3name      Gerenciamento de 8dot3name
behavior        Controla o comportamento do sistema de arquivos
dax             Gerenciamento de volume Dax
dirty          Gerencia o bit sujo de volume
file           Comandos específicos de arquivo
fsInfo         Informações do sistema de arquivos
hardlink       Gerenciamento de link físico
objectID       Gerenciamento de IDs de objetos
quota          Gerenciamento de cotas
repair         Gerenciamento de autorrecuperação
reparsePoint   Gerenciamento de ponto de nova análise
storageReserve Gerenciamento de Reserva de Armazenamento
resource       Administração do Gerenciador de Recursos de Transação
sparse         Controle de arquivos esparsos
tiering        Gerenciamento de propriedades de camadas de armazenamento
transaction    Gerenciamento de transação
usn            Gerenciamento de USN
volume         Gerenciamento de volume
wim            Gerenciamento de hospedagem do Wim transparente
```

**Fonte:** elaborado pelos autores, 2025.

## Criando um arquivo com **fsutil**

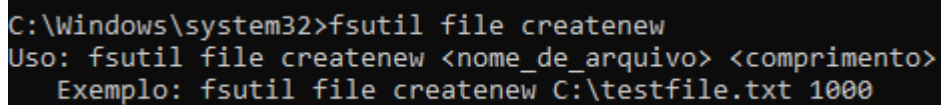
**Imagem 9** – Utilizando o fsutil



```
C:\Users\wg761>fsutil file createnew video.txt 1000
Arquivo C:\Users\wg761\video.txt criado
```

**Fonte:** elaborado pelos autores, 2025.

**Imagem 10** – Utilizando o fsutil



```
C:\Windows\system32>fsutil file createnew
Uso: fsutil file createnew <nome_de_arquivo> <comprimento>
Exemplo: fsutil file createnew C:\testfile.txt 1000
```

**Fonte:** elaborado pelos autores, 2025.

Vale ressaltar que para a criação de um arquivo você deve indicar o caminho pelo qual o arquivo irá estar no meu caso foi pelo C:

### 3.3 Android

O acesso ao Shell do Android foi feito pelo app Termux no próprio dispositivo.

Comando `ls -l`

Como vimos anteriormente esse comando serve pra listar os diretórios e arquivos presentes na máquina.

Imagem 11 – Utilizando o Android

```
~ $ ls -l
total 4
drwx----- 2 u0_a401 u0_a401 3452 Oct 12 20:36 teste
```

Comando `chmod`

O comando `chmod` serve pra alterar as propriedades de uso de arquivos, ou seja suas permissões.

Imagem 12 – Utilizando o chmod.

```
~ $ chmod -x teste
~ $ ls -l
total 4
drw----- 2 u0_a401 u0_a401 3452 Oct 12 20:36 teste
~ $
```

Nesse caso alteramos que o arquivo teste não tem a permissão de ser executado com o `x`.

#### **4. Comparação e Análise Crítica.**

Com base nos comandos e informações realizados e apresentados de acordo com cada sistema operacional, ambos trazem a mesma ideia de ser realizado pelo CLI e entretanto o Linux e Android se propuseram a ser mais fáceis de gerenciar pela sua familiaridade enquanto o Windows se tornou difícil de mexer.

##### **4.1 Linux**

- Filosofia: O Ubuntu adota um modelo simples e direto baseado em permissões numéricas (rwx) e propriedade por usuário e grupo
- Controle: é altamente flexível e eficiente em ambiente multiusuário.

##### **4.2 Windows**

- Filosofia: Utiliza ACLs e grupos, permitindo definir permissões específicas para múltiplos usuários com herança e negação.
- Controle: Ideal para ambientes corporativos com políticas complexas de segurança.

##### **4.3 Android**

- Filosofia: Segurança sendo prioridade por conta de cada aplicativo rodar num UID único.
- Controle: O usuário concede permissões explícitas (como acesso à câmera ou localização), mas não há arquivos do sistema.

## **5 CONCLUSÃO**

Com o uso e entendimento do Sistema de Arquivos (SA), pode-se compreender a manipulação de arquivos de acordo com cada sistema operacional e colocar em prática.

Concluindo que este relatório conseguiu cumprir o que lhe foi proposto que foi a manipulação e entendimento de permissões de arquivos.

## **6 AUTOAVALIAÇÃO**

O grupo foi muito dedicado e colaborativo com este projeto, oque lhe foi possível chegar à esse resultado que para ambos foi extremamente satisfatório. Entretanto o grupo sofreu com algumas dificuldades:

- A Inclusão de permissões nos três sistemas operacionais
- A criação e manipulação dos arquivos
- Entender os conceitos de IO, CI etc..

Mas apesar das dificuldades temos ciência que demos o melhor e isso foi concluído com êxito.

## 7 REFERÊNCIAS

**MAILCHIMP.** *File management.* Disponível em: <https://mailchimp.com/pt-br/resources/file-management/>. Acesso em: 12 out. 2025.

**DOCINT.** *O que é gerenciamento de permissões para usuários administrativos.* Disponível em: <https://www.docint.com.br/glossario/o-que-e-gerenciamento-de-permissoes-para-usuarios-administrativos/#:~:text=A%20import%C3%A2ncia%20do%20gerenciamento%20de,a%20LGPD%20e%20a%20GDPR>. Acesso em: 12 out. 2025.

**DOCUWARE.** *File management: simplifying data organisation.* Disponível em: <https://start.docuware.com/en-gb/blog/file-management-simplifying-data-organisation/#:~:text=O%20gerenciamento%20de%20arquivos%20%C3%A9%20essencial%20para%20que%20as%20empresas,otimizados%20e%20aumento%20de%20produtividade>. Acesso em: 12 out. 2025.

**IBM.** *Security directory server: security directory access control list (ACL).* Disponível em: <https://www.ibm.com/docs/pt-br/sdse/6.4.0?topic=directory-security>. Acesso em: 12 out. 2025.

**IBM.** *Security Verify Access: policy definition and application security.* Disponível em: <https://www.ibm.com/docs/pt-br/sva/11.0.0?topic=policy-definition-application-security>. Acesso em: 12 out. 2025.

**YOUTUBE.** *Comando ICACLS – Gerenciamento de permissões no Windows.* Disponível em: <https://youtu.be/sq6pd18X63Q>. Acesso em: 12 out. 2025.

**RED HAT.** *DU vs DF: understanding disk usage commands in Linux.* Disponível em: <https://www.redhat.com/en/blog/du-vs-df#:~:text=df%20vs.,Baixe%20agora%20mesmo%20gratuitamente>. Acesso em: 12 out. 2025.

**YOUTUBE.** *Gerenciamento de arquivos no Linux.* Disponível em: <https://youtu.be/5R-l2p3tu6k>. Acesso em: 12 out. 2025.

**YOUTUBE.** *Uso do comando FSUTIL no Windows.* Disponível em: <https://youtu.be/WzmyGB3eFFs>. Acesso em: 12 out. 2025.



